Sentinel: Attention Probing of Proxy Models for LLM Context Compression with an Understanding Perspective

Yong Zhang[1], **Yanwen Huang**[1,2], **Ning Cheng**[1,*],
Yang Guo[1], **Yun Zhu**[1], **Yanmeng Wang**[1], **Shaojun Wang**[1], **Jing Xiao**[1],
[1] Ping An Technology (Shenzhen) Co., Ltd., China
[2] University of Electronic Science and Technology of China
zhangyong203@pingan.com.cn

Abstract

Retrieval-augmented generation (RAG) enhances large language models (LLMs) with external context, but retrieved passages are often lengthy, noisy, or exceed input limits. Existing compression methods typically require supervised training of dedicated compression models, increasing cost and reducing portability. We propose **Sentinel**, a lightweight sentence-level compression framework that reframes context filtering as an *attention-based understanding task*. Rather than training a compression model, Sentinel probes decoder attention from an off-the-shelf 0.5B proxy LLM using a lightweight classifier to identify sentence relevance. Empirically, we find that *query-context relevance estimation is consistent across model scales*, with 0.5B proxies closely matching the behaviors of larger models. On the LongBench benchmark, Sentinel achieves up to 5× compression while matching the QA performance of 7B-scale compression systems. Our results suggest that probing native attention signals enables fast, effective, and question-aware context compression.[1]

1 Introduction

Large language models (LLMs) have achieved impressive performance across open-domain question answering, reasoning, and dialogue tasks (Brown et al., 2020; OpenAI, 2024). To scale their capabilities to knowledge-intensive applications, Retrieval-Augmented Generation (RAG) has emerged as a powerful paradigm that augments model inputs with retrieved evidence from external corpora (Lewis et al., 2020; Guu et al., 2020; Shi et al., 2024). However, long retrieved contexts are often noisy, redundant, or exceed model input limits, making **context compression** essential for both efficiency and effectiveness (Liu et al., 2024; Yoran et al., 2024).

This challenge has motivated a shift from coarse document-level reranking (Karpukhin et al., 2020) to fine-grained context compression, broadly categorized into token-level and sentence-level selection strategies. Token-based methods (e.g., LLMLingua 1/2 (Jiang et al., 2023; Pan et al., 2024), QGC (Cao et al., 2024)) estimate token importance via perplexity or query-aware signals but often fragment discourse coherence. Sentence-level approaches (e.g., RECOMP, EXIT (Xu et al., 2024; Hwang et al., 2024)) preserve syntactic structure by selecting full sentences, yet typically require generator feedback or task-specific tuning. Despite progress, existing methods remain tightly coupled to supervision or generation signals—highlighting the need for a lightweight alternative.

Among these challenges, a growing body of work suggests that LLMs inherently reveal internal semantic and relevance signals during inference. Specifically, decoder attention patterns have been shown to capture factual attribution and grounding behaviors (Wu et al., 2024; Huang et al., 2025), while final hidden states under specialized prompts can serve as effective compressed semantic embeddings, as demonstrated in PromptEOL (Jiang et al., 2024b). These findings point to an emerging consensus: *LLMs naturally aggregate context understanding into their final inference steps*, providing an opportunity for lightweight signal extraction without explicit generation.

Building on these findings, we hypothesize that **query-context understanding tends to be stable across model scales**, even as generation capabilities differ. Smaller models may exhibit attention patterns that align closely with those of larger LLMs, despite limited generation capacity. This suggests a promising direction: instead of relying on full-scale generation or reranking, we can decode relevance signals directly from native attention behaviors in compact models. We adopt this perspective to design a lightweight and efficient

* Corresponding author
[1] Our code is available at https://github.com/yzhangchuck/Sentinel.

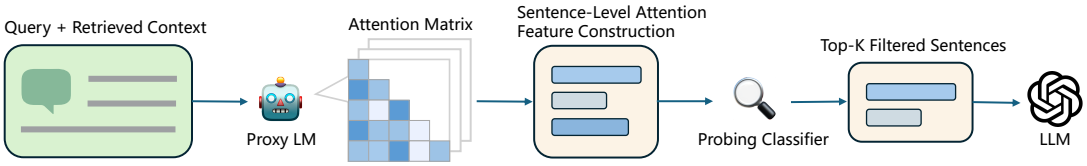


Figure 1: **Sentinel Framework Overview.** Given a query–context pair, an off-the-shelf proxy model provides final-token decoder attention. A probing classifier interprets sentence-level attention features to select relevant context, enabling lightweight and model-agnostic compression for downstream LLMs.

approach to context compression.

However, existing methods often use attention in shallow or heuristic ways. Many rely on raw thresholding over decoder attention, which tends to be noisy, brittle, and poorly aligned with sentence-level semantics (Wang et al., 2024; Fang et al., 2025). These approaches fall short of providing a lightweight and interpretable compression mechanism that can robustly leverage native model behavior.

In this work, we propose **Sentinel**, a lightweight and model-agnostic framework for sentence-level context compression based on attention probing. Given a query–context pair, Sentinel uses an off-the-shelf 0.5B-scale proxy LLM to extract multi-layer decoder attention directed toward input sentences. Instead of training a dedicated compression model, we treat compression as a probing task: aggregating attention features across heads and layers, and using a simple logistic regression classifier to predict sentence relevance. This design eliminates the need to train a task-specific compression model enabling efficient, interpretable, and plug-and-play compression.

Empirically, Sentinel achieves up to 5× input compression on LongBench (Bai et al., 2024), while matching the QA performance of 7B-scale compression systems using only a 0.5B proxy model. It generalizes effectively across QA tasks, languages, and LLM backbones, without requiring generator supervision or prompt-specific tuning.

Our contributions are as follows:

- We reframe context compression as an **attention-based understanding task**, and implement it via a probing classifier—without training a dedicated compression model.
- We propose **Sentinel**, a lightweight sentence-level compression framework that probes native attention signals from an off-the-shelf 0.5B proxy model to identify query-relevant context for compression.
- We empirically observe that **query-context relevance estimation remains stable across model scales**, enabling compact proxies to approximate large-model compression behavior.
- On LongBench, Sentinel achieves up to 5× input compression while matching the QA performance of 7B-scale compression systems across both English and Chinese tasks.

2 Methodology

We propose **Sentinel**, a lightweight framework that probes native attention behaviors in small proxy models to identify query-relevant sentences. Rather than training a compression model end-to-end, we decode attention-based signals already embedded in the model's inference dynamics. Our pipeline consists of three stages: attention feature extraction, probing classifier training, and integration with downstream LLMs.

2.1 Task Formulation

Given a query q and a retrieved context $C = \{s_1, s_2, \ldots, s_n\}$ composed of sentences, our goal is to select a subset $C' \subseteq C$ that retains essential information for answering q.

We frame this as a probing task: for each sentence s_i, we train a binary classifier to predict a relevance label $y_i \in \{0, 1\}$ based solely on attention signals from a compact proxy model. At inference time, we use the predicted probability as a soft relevance score for sentence selection.

2.2 Attention-Based Utility Estimation

The core of Sentinel is a feature extractor that leverages decoder attention to estimate sentence utility. Given the input sequence:

$$[q] \oplus [s_1] \oplus [s_2] \oplus \cdots \oplus [s_n]$$

we feed it into a small decoder-only language model with instruction-following capability. To encourage semantic compression at the final position, we apply a prompt template that requests a

one-word answer following the context and query, similar in spirit to PromptEOL (Jiang et al., 2024b). We extract the attention tensor $\mathbf{A} \in \mathbb{R}^{L \times H \times T}$ from the final decoder token, capturing attention scores across layers, heads, and input tokens.

Why attention reflects utility. Decoder attention has been shown to reflect alignment and attribution signals (Huang et al., 2025). In particular, the attention received by the final token often encodes which input segments are most relevant for generation. From an information-theoretic perspective, Barbero et al. (2024) show that decoder-only models compress prior context into the final token representation — *over-squashing*.

Feature representation. For each sentence s_i, we compute a set of attention-based features derived from the attention received by its tokens from the final decoder token. Specifically, we extract the attention weights from each layer and head directed toward context tokens, and normalize them by the total attention mass over the context span. This normalization removes the influence of non-context elements such as the query or prompt.

We then average the normalized attention weights over the tokens in s_i, independently for each attention head and layer, yielding a feature vector $\mathbf{v}_i \in \mathbb{R}^{LH}$, where L is the number of decoder layers and H is the number of attention heads. Each element of $\mathbf{v}_i$ reflects the relative contribution of s_i as measured by a specific attention channel.

2.3 Probing Classifier for Sentence Relevance

To decode relevance from attention features, we train a lightweight logistic regression (LR) probe. The probe computes:

$$\hat{y}_i = \sigma(\mathbf{w}^\top \mathbf{v}_i + b),$$

where σ is the sigmoid function. The model outputs a probability score for each sentence, which is used directly for sentence ranking.

2.4 Weak Supervision and Robust Probing

To effectively probe the model's internal relevance behavior, we train the classifier using weak supervision from QA datasets, combined with retrieval filtering and robustness enhancements.

2.5 Probing Data Construction

We construct the initial training data from widely used QA datasets, covering both single-hop and multi-hop question answering scenarios. Specifically, we use examples from SQuAD, Natural Questions (single-hop), and HotpotQA (multi-hop), where answer spans are annotated in retrieved contexts. For each QA example, sentences containing the gold answer span are labeled as positive, while all other sentences are labeled as negative. This weak supervision allows us to build large-scale training data without requiring manual annotation of sentence relevance, and ensures that the classifier is exposed to both simple factual questions and complex multi-hop reasoning patterns.

2.5.1 Selecting Context-Reliant Samples

To purify supervision, we retain only QA examples that require retrieved context for correct answering. Specifically, We retain only QA examples where the model fails to answer correctly without the retrieved context but succeeds when the context is provided. This filtering conceptually echoes prior work that probes model behavior via intervention-based output changes (Meng et al., 2022). This criterion ensures that the positive sentences truly contribute critical information needed for answering, and reduces contamination from internal model memorization or hallucinated knowledge. By filtering for retrieval-dependency, we focus training on cases where relevance decoding from context is essential, aligning with the goal of context understanding rather than internal recall.

2.5.2 Robustness via Sentence Shuffling

To mitigate positional biases (Liu et al., 2024), especially common in multi-document retrieval settings, we apply sentence shuffling during training by randomly permuting sentence order within each passage. This simple perturbation encourages the classifier to rely on semantic relevance rather than fixed positions, improving generalization to real-world RAG inputs with noisy or varied structure.

2.6 Inference via Attention Probing

At inference time, given a query–context pair (q, C), we run the proxy model with a fixed QA-style prompt, extract final-token decoder attention, and compute sentence-level attention features. The probing classifier assigns a relevance score to each sentence, and the top-ranked subset is selected to fit the length budget. This compressed context is passed to the downstream LLM for generation.

Since Sentinel operates solely on proxy model attention, it is **model-agnostic** and can be inte-

grated into any RAG pipeline without modifying or fine-tuning the target LLM.

3 Experiments

Datasets We evaluate our method on the English subset of LongBench (Bai et al., 2024), which covers six task categories: *Single-Document QA*, *Multi-Document QA*, *Summarization*, *Few-shot Reasoning*, *Synthetic Retrieval*, and *Code Completion*.

To ensure comparability with prior work (e.g., GPT-3.5-Turbo baselines), we include all original tasks in comparison tables. However, for our Qwen-based experiments, we exclude **LCC** and **PassageCount** as their input formats and task goals conflict with context compression (see Appendix A). In our Qwen-based tables, we annotate the **Synth.** and **Code** columns with an asterisk (*) to indicate modified category composition.

Probing Data We construct our training set from 3,000 QA examples sampled from NewsQA (50%), SQuAD (20%), and HotpotQA (30%), covering both single-hop and multi-hop reasoning tasks. For each QA example, we extract one positive sentence supporting the gold answer span, and one negative sentence from the same passage, resulting in 6,000 sentence-level training instances.

In NewsQA, 30.1% of examples contain 0–500 tokens (tokenized with Qwen-2.5) and 69.9% have 500–1,000 tokens. In SQuAD, 99.3% fall within 0–500 tokens. For HotpotQA, we restrict all examples to 0–500 tokens by limiting unrelated content.

Each context is segmented into sentences using spaCy's `sentencizer`. To extract features, we apply a QA-style prompt to each example and collect decoder attention from the final token. The prompt format is:

```
Given the following information: {context}
Answer the following question based on the
given information with one or few words:
{question}
Answer:
```

Decoder attention weights from the final token to each sentence are aggregated to form fixed-length feature vectors for classification.

Context-Reliant Sample Selection To improve label quality, we retain only examples where the context is necessary for correct answering. For NewsQA and SQuAD, we keep examples where the memory-based answer is incorrect and the context-based answer is correct, both judged by EM. For HotpotQA, we retain only samples with memory F1 ≤ 0.2 and context F1 ≥ 0.5.

Probing Classifier Training We train a logistic regression (LR) model on attention-derived features, using 5-fold cross-validation with balanced accuracy as the scoring metric. We perform grid search over regularization strengths $C \in \{0.01, 0.1, 1.0, 10.0, 100.0\}$, and use the liblinear solver with ℓ_2 regularization, class-balanced weighting, and a maximum of 2,000 iterations. The best model is selected based on AUC on the validation set.

Compression Strategy We implement a length-controlled compression strategy based on the target LLM's tokenizer. For a given context $x = s_1, s_2, \ldots, s_n$, our classifier scores each sentence, and we select a subset that meets one of the following constraints:

- **Token-length budget:** Retain top-ranked sentences until their total token count (measured by the target model's tokenizer) reaches a fixed budget B (e.g., 2000 tokens).
- **Token-ratio constraint:** Retain top-ranked sentences whose cumulative token count does not exceed a fraction τ (e.g., 0.1 to 0.5) of the original context's tokenized length.

In both cases, selected sentences are concatenated in their original order to form the compressed input.

Proxy Model Setup In our main experiments, we adopt **Sentinel** with `Qwen-2.5-0.5B-Instruct` as the default proxy model for extracting attention-based features and performing sentence relevance classification. Unless otherwise specified, all reported results use a chunk size of 1024 tokens, where the retrieved context is segmented into non-overlapping token blocks of up to 1024 tokens before being passed to the proxy model.

Evaluation Models Following the LongBench and LLMLingua setup, we use `ChatGPT (gpt-3.5-turbo)` as the primary model for QA evaluation. To assess the generality of our method, we also experiment with `Qwen-2.5-7B-Instruct` in our main results. All evaluations follow the LongBench prompt and decoding setup (Bai et al., 2024), as detailed in Appendix J.

Methods	LongBench (GPT-3.5-Turbo, 2000-token constraint)							Compression Stats	
	SingleDoc	MultiDoc	Summ.	FewShot	Synth.	Code	**AVG**	Tokens	$1/\tau$
Selective-Context (LLaMA-2-7B-Chat)	16.2	34.8	24.4	15.7	8.4	49.2	24.8	1,925	5x
LLMLingua (LLaMA-2-7B-Chat)	22.4	32.1	24.5	61.2	10.4	56.8	34.6	1,950	5x
LLMLingua-2	29.8	33.1	25.3	66.4	21.3	58.9	39.1	1,954	5x
LongLLMLingua (LLaMA-2-7B-Chat)	39.0	42.2	**27.4**	69.3	**53.8**	56.6	48.0	1,809	6x
CPC (Mistral-7B-Instruct-v0.2)	**42.6**	**48.6**	23.7	69.4	52.8	**60.0**	**49.5**	1,844	5x
Sentinel (Qwen-2.5-0.5B-Instruct)	40.1	47.4	25.8	**69.9**	46.3	58.0	47.89	1,885	5x
Sentinel (Qwen-2.5-1.5B-Instruct)	40.6	48.1	26.0	69.1	49.0	57.6	48.4	1,883	5x
Original Prompt	39.7	38.7	26.5	67.0	37.8	54.2	44.0	10,295	-

Table 1: Performance on LongBench using GPT-3.5-Turbo as the inference model. Best results are in **bold**, second-best are underlined.

Methods	LongBench-En (Qwen-2.5-7B-Instruct, 2000-token constraint)							LongBench-Zh (Qwen-2.5-7B-Instruct, 2000-token constraint)						
	SingleDoc	MultiDoc	Summ.	FewShot	*Synth.	*Code	En-AVG	SingleDoc	MultiDoc	Summ.	FewShot	Synth.	Zh-AVG	Overall AVG
Empty Context	10.74	19.97	13.52	40.1	2.5	56.0	23.81	15.48	12.25	10.06	18.5	4.38	12.13	19.78
Random	26.54	28.03	23.83	62.5	18.25	62.12	36.88	42.37	17.64	13.93	19.25	35.0	25.64	36.33
Raw Attention (0.5B)	33.11	38.39	24.22	63.27	77.67	62.14	49.8	56.24	20.99	13.61	**44.50**	72.76	41.62	48.47
Sentinel (0.5B)	37.11	44.98	25.02	64.88	85.54	63.04	53.43	59.18	24.15	13.19	43.00	74.68	42.84	51.23
Sentinel (1.5B)	38.47	44.95	25.03	65.32	90.50	62.16	54.40	58.88	25.52	13.54	42.17	**83.57**	44.74	52.48
Sentinel (3B)	**39.06**	**45.44**	25.21	66.00	86.46	62.93	54.18	59.10	**25.83**	13.86	43.00	77.03	43.76	52.04
Original Prompt	37.88	40.31	**25.32**	**69.21**	**98.58**	**66.74**	**56.34**	**60.60**	20.67	**15.03**	43.25	**86.93**	**45.30**	**54.05**

Table 2: Joint performance on LongBench-En and LongBench-Zh using Qwen-2.5-7B-Instruct. Best results are in **bold**, second-best are underlined.

Baselines We compare Sentinel against a range of context compression baselines, including token-level methods (LLMLingua-1/2 (Jiang et al., 2023; Pan et al., 2024)), sentence-level approaches (Selective-Context (Li et al., 2023), CPC (Liskavets et al., 2024), LongLLMLingua (Jiang et al., 2024a)), and attention-based heuristics such as Raw Attention (Wang et al., 2024; Fang et al., 2025). We also include non-learning baselines including Random Selection and Empty Context. Full descriptions of these baselines are provided in Appendix B.

Metrics We follow the LongBench evaluation protocol and adopt task-specific metrics for each task category: QA-F1 for Single-Document QA, Multi-Document QA, and Few-shot Reasoning; ROUGE-L for Summarization; classification accuracy for Synthetic Retrieval. All metrics are computed using the official evaluation scripts.

3.1 Results on LongBench

We evaluate Sentinel across two settings: (1) English tasks using GPT-3.5-Turbo as the inference model, and (2) both English and Chinese tasks using Qwen-2.5-7B-Instruct. All results are reported under a 2,000-token input constraint with chunk size 1024 unless otherwise noted. Although the probing classifier is trained on external QA datasets, it generalizes effectively to downstream LongBench tasks.

English Evaluation with GPT-3.5-Turbo Table 1 shows that Sentinel, with a 0.5B proxy, performs strongly across all English LongBench categories, outperforming task-agnostic baselines like LLMLingua-1/2 and matching 7B-scale systems like CPC and LongLLMLingua. This suggests that small proxies suffice for effective sentence selection. On Chinese tasks, Sentinel also outperforms LLMLingua under tight budgets when evaluated with GPT-3.5-Turbo (Appendix E).

English Evaluation with Qwen-2.5-7B In Table 2, we report results on the English subset of LongBench using Qwen-2.5-7B-Instruct. Sentinel consistently outperforms Raw Attention, Random, and Empty Context baselines across all task types. Notably, on Single-Doc and Multi-Doc QA, Sentinel even surpasses the Original Prompt despite using a 5$\times$ shorter input, confirming its ability to distill high-relevance content.

However, on Summarization, FewShot, and Code tasks, Sentinel underperforms the Original Prompt. This is likely due to the loss of global structure: summarization benefits from full-passage context, few-shot tasks rely on intact example formatting, and code inputs are sensitive to line breaks, which our sentence segmentation does not preserve.

Chinese Evaluation with Qwen-2.5-7B On the Chinese subset of LongBench, Sentinel demonstrates competitive performance, generally outperforming Raw Attention across tasks. It achieves

strong results on Single-Doc and notably surpasses the Original Prompt on Multi-Doc QA. While its performance on FewShot and Summarization remains lower than the Original, Sentinel performs comparably on the Synth task, suggesting robustness to long-context scenarios despite minor formatting disruptions.

Overall, these results demonstrate that Sentinel generalizes well to multilingual settings and remains effective in compressing high-relevance content across languages.

Learning vs. Raw Attention Sentinel (0.5B) outperforms Raw Attention (0.5B) on most tasks, especially Single-Doc, Multi-Doc QA, and Synthetic tasks. Improvements are consistent in English and smaller but generally positive in Chinese, supporting the value of learning explicit relevance over using raw attention scores.

3.2 Proxy Model Size Robustness

Our framework is based on the hypothesis that query-context relevance, as reflected in attention, is relatively stable across proxy model scales. To evaluate this, we compare three Qwen-2.5 models (0.5B, 1.5B, 3B) under the same training setup, evaluated with 1024-token chunks and a 2,000-token input budget. As shown in Table 2, performance remains relatively stable across scales, with average F1 varying by less than 2 points. The 0.5B model already achieves competitive results at a much lower computational cost, supporting the efficiency of using smaller proxy models.

To further assess alignment across models, we compute pairwise sentence-level overlap in selected sentences. Sentence-level overlap increases with budget, ranging from 0.63–0.70 at 1000 tokens to 0.74–0.78 at 3000 tokens, suggesting consistent relevance estimation across proxy model scales. Full results are provided in Appendix F.

These findings confirm that attention-based relevance estimation is stable across model scales. A 0.5B proxy can closely approximate the sentence selection behavior of larger models, enabling accurate and efficient compression.

3.3 Ablation

We ablate three aspects of our method: attention feature design, chunk size, and sentence selection strategy. Unless specified, all experiments use `Qwen-2.5-7B-Instruct` for generation and are evaluated on LongBench.

Feature	HotpotQA	SQuAD	NewsQA	Overall AUC	Overall AVG
All Layers	0.9228	0.9987	0.9838	**0.9700**	51.23
Selected	0.9171	0.9943	0.9832	0.9662	50.20
Last Layer	0.8606	0.9538	0.9588	0.9121	49.04

Table 3: AUC comparison of different attention feature extraction strategies and their effectiveness on Qwen-2.5-7B-Instruct compressed evaluation.

3.3.1 Attention Feature Ablations

We evaluate three attention-based feature construction strategies on Qwen-2.5-0.5B-Instruct.

1. **All Layers**: Aggregate attention scores from all decoder layers.
2. **Last Layer**: Use only the attention scores from the final decoder layer.
3. **Selected**: Use mRMR (Ding and Peng, 2005) to select a compact set of attention heads, limited to no more than those in one decoder layer. See Appendix C for details.

As shown in Table 3, the All Layers strategy achieves the highest AUC and downstream performance, while the Selected variant offers a strong trade-off between compactness and accuracy.

3.3.2 Chunk Size Variants

We evaluate Sentinel under varying chunk sizes (512, 1024, 2048, 4096), all constrained to a 2000-token budget during inference. Larger chunks allow attention to span broader contiguous context, while reducing the number of input segments. As shown in Figure 2 (top), Sentinel consistently outperforms the Raw Attention baseline across all chunk sizes.

Although the proxy was trained on sequences shorter than 1024 tokens, performance continues to improve with larger chunk sizes, reaching the best result at 4096 tokens. This improvement may stem from the model's ability to attend over a wider context window during inference, which facilitates more effective context compression. Full task-level results are provided in Appendix D.

3.3.3 Compression Ratio Variants

We further evaluate robustness under compression constraints $\tau \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$, where lower τ denotes more aggressive context pruning. As shown in Figure 2 (bottom), Raw attention degrades sharply under tighter constraints, especially when $\tau < 0.3$. In contrast, Sentinel maintains stable performance across all compression levels, with only mild degradation even at $\tau = 0.1$.

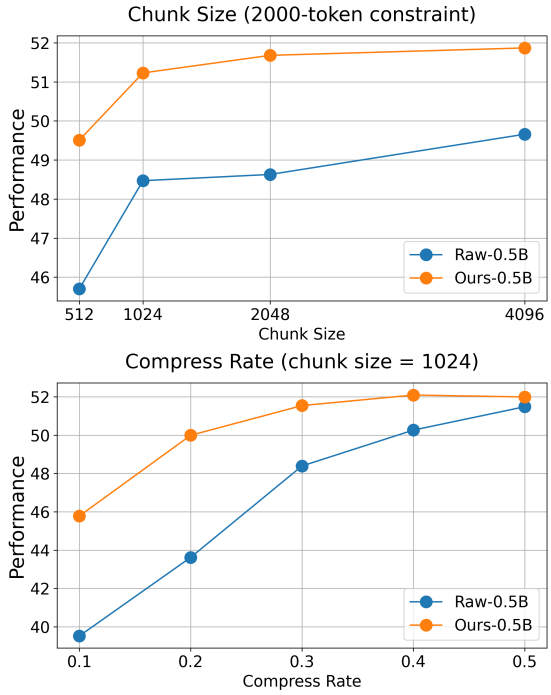


Figure 2: Ablation results on `Qwen-2.5-7B-Instruct` with 0.5B proxy. **Top:** Chunk size ablation under a 2000-token constraint. **Bottom:** Compression ratio ablation at chunk size 1024.

These results highlight Sentinel's robustness in low-resource scenarios and its ability to extract semantically rich signals under strong token budgets. Full task-level results are available in Appendix D.

3.3.4 Latency and Inference Efficiency

We evaluate end-to-end inference latency across different Sentinel configurations, focusing on the effects of chunk size and attention feature design. Table 4 reports average and median latency per sample on the English LongBench dataset, measured on a single A100 GPU.[2]

With a chunk size of 1024 and All Layers attention features, Sentinel achieves 1.13× speedup over LLMLingua-2 while reaching 51.23 F1. Increasing the chunk size to 2048 yields both higher accuracy (51.68 F1) and slightly faster inference (0.65s avg), demonstrating the benefit of longer-span attention at inference time. Interestingly, using even larger chunks (4096) leads to the best overall accuracy (51.87 F1), but latency returns to 0.78s—on par with LLMLingua-2. This is due to increased GPU memory pressure from computing longer-range attention, which becomes the primary bottleneck at this scale.

[2]We monkey-patch the model to extract only the final-token attention used by our method, replacing other activations with `None` to reduce overhead.

To further improve runtime, we evaluate SENTINEL (SELECTED), which uses compact mRMR-selected features. At chunk size 1024, this variant reduces latency to 0.60s (1.30×) with only minor performance degradation (50.20 F1), offering an efficient alternative for low-latency scenarios.

4 Analysis and Discussion

4.1 Probing Feature Distributions across Datasets and Scales

We visualize probing features via t-SNE (Appendix G). Positive/negative samples are well-separated in SQuAD and NewsQA, but overlap in HotpotQA—highlighting the challenge of multi-hop compression. The feature distributions remain consistent across proxy scales, reinforcing the stability of attention-based relevance. We also vary the number of probing examples (Appendix **??**) and observe stable performance across training sizes.

4.2 Attention Behavior and Sentence Selection

To better understand how Sentinel decodes sentence-level relevance, we visualize attention behavior on an example from `2WikiMultihopQA`, a multi-hop QA task in LongBench. The input is processed using the `Qwen-2.5-0.5B-Instruct` proxy model with a chunk size of 1024.

We compare three strategies for visualizing sentence importance: (1) **Token-Level Attention**, which directly displays the attention weights from the final decoder token to each input token; (2) **Averaged Sentence Attention** (corresponding to our Raw Attention baseline), which computes sentence-level scores by averaging attention weights over each sentence's tokens; and (3) **Sentinel Probing Prediction**, which uses a trained classifier to output sentence relevance probabilities based on attention-derived features.

As shown in Appendix I, raw token-level attention exhibits a strong attention sink effect (Bondarenko et al., 2021; Son et al., 2024), with a large portion of weights concentrated on the final token in the input. As a result, sentence-level averaging (i.e., the Raw Attention baseline) is heavily skewed and often fails to reflect true semantic relevance. While attention distributions do encode useful signals, they are noisy and unstructured. In contrast, Sentinel's probing-based classifier avoids attention sink and more effectively decodes relevance patterns embedded in the model's behavior.

Compression as Understanding Unlike prior methods that require training a dedicated com-

Method	Chunk Size	Proxy Model	Avg. Time (s) ↓	Med. Time (s) ↓	Speedup vs. LLMLingua-2 ↑	Overall-AVG
LLMLingua-2 (trained)	512	XLM-RoBERTa-Large (561M)	0.78	0.70	1.00×	32.65
Raw Attention	512	Qwen-2.5-0.5B-Instruct (494M)	1.01	0.84	0.77×	45.70
Raw Attention	1024	Qwen-2.5-0.5B-Instruct (494M)	0.65	0.54	1.20×	48.47
Sentinel (ours)	512	Qwen-2.5-0.5B-Instruct (494M)	1.02	0.84	0.76×	49.51
Sentinel (ours) selected	512	Qwen-2.5-0.5B-Instruct (494M)	0.85	0.70	1.09×	46.30
Sentinel (ours)	1024	Qwen-2.5-0.5B-Instruct (494M)	0.69	0.57	1.13×	51.23
Sentinel (ours) selected	1024	Qwen-2.5-0.5B-Instruct (494M)	0.60	0.49	1.30×	50.20
Sentinel (ours)	2048	Qwen-2.5-0.5B-Instruct (494M)	0.65	0.51	1.20×	51.68
Sentinel (ours)	4096	Qwen-2.5-0.5B-Instruct (494M)	0.78	0.64	1.00×	**51.87**

Table 4: Inference latency per QA sample on the full LongBench dataset (lower is better). LLMLingua-2 is trained for token-level compression and limited to 512-token chunks. Sentinel uses a smaller, untrained decoder-only proxy and supports larger chunk sizes for improved efficiency. Speedup is relative to LLMLingua-2 (chunk size 512). **Overall-AVG** denotes average accuracy across all chunk settings per method.

pression model with large-scale labeled data, we frame compression as a pure understanding problem: identifying which parts of the context are relevant to the question. Crucially, we do not train a compression model directly—instead, we probe the attention behaviors of a small proxy LLM to extract relevance signals already embedded in its internal computation. This design leverages the model's native understanding capabilities, rather than its generation capacity. Our experiments show that such relevance behaviors are stable across model scales: even a 0.5B proxy yields similar sentence-level decisions as larger models. Although the probing classifier is trained on a small set of external QA data (3,000 examples), it generalizes effectively to the diverse downstream tasks in LongBench. This enables a lightweight and interpretable framework for query-aware compression, with no need for task-specific tuning or large-scale supervision.

5 Related Work

Token-Level Compression Token-level methods aim to prune irrelevant or redundant content at fine granularity. LLMLingua 1/2 (Jiang et al., 2023; Pan et al., 2024) estimates token importance via self-information and token selection distillation using small LMs. QGC (Cao et al., 2024) performs query-guided compression by aligning token representations with the query through pooling and embedding refinement. While these approaches achieve high compression ratios, they often fragment discourse coherence.

Sentence-Level Compression Sentence-level compression preserves semantic units by selecting full sentences rather than tokens. Extractive methods such as RECOMP (Xu et al., 2024) and EXIT (Hwang et al., 2024) formulate compression as a binary classification task, supervised by generator feedback or contrastive signals. CPC (Liskavets et al., 2024), in contrast, learns a sentence encoder to rank sentences by query relevance, optimizing a retrieval-style objective. Structure-enhanced methods like Refiner (Li et al., 2024) and FineFilter (Zhang et al., 2025) further reorganize or rerank selected content to support multi-hop reasoning and long-context understanding. While these approaches are effective, they often require large-scale supervision or task-specific tuning, which limits their adaptability across tasks and models.

Attention-Based Compression Recent work explores leveraging decoder attention as a native relevance signal. QUITO (Wang et al., 2024) introduces a trigger token to guide attention-based token scoring, while ATTENTIONRAG (Fang et al., 2025) reformulates QA as masked prediction and uses attention to prune low-utility sentences. However, these methods often rely on prompt engineering or direct thresholding of raw attention, limiting generality and robustness.

Our Approach In contrast to prior work, we propose **Sentinel**, a lightweight and model-agnostic framework that probes native attention signals from small proxy models to predict sentence relevance. Rather than training a compression model or relying on raw scoring mechanisms, Sentinel treats compression as an understanding problem—decoding which parts of the input are internally attended to during question answering. By empirically validating the stability of attention-based relevance across model scales, our approach enables efficient and interpretable compression without generation supervision or task-specific training.

6 Conclusion

We present **Sentinel**, a lightweight and interpretable sentence-level compression framework that probes multi-layer attention from an off-the-shelf 0.5B proxy model to decode sentence rele-

vance—without supervision or task-specific tuning. On LongBench, Sentinel achieves up to 5× compression while matching the QA performance of 7B-scale systems across English and Chinese tasks. These results show that attention probing offers an efficient alternative to supervised compression, and that small models can effectively support context understanding in large language systems.

Limitations

Format Sensitivity in FewShot and Code Tasks. Our approach uses generic sentence segmentation and does not account for task-specific structural constraints. As a result, performance on FewShot and Code tasks may degrade due to formatting disruption—such as broken input–output pairs in FewShot prompts or misaligned line boundaries in code. These tasks are inherently sensitive to layout and structure, which current compression does not explicitly preserve.

Proxy Model Scope. All probing experiments use Qwen-2.5-Instruct models as the proxy. While Qwen offers strong alignment and open availability, we have not tested cross-architecture generalization to other families such as LLaMA, Mistral, or GPT-based models. Future work could explore whether attention patterns are equally probe-able across diverse architectures.

Limited Evaluation Backbones. Our evaluation is conducted on two decoder LLMs—GPT-3.5-Turbo and Qwen-2.5-7B-Instruct. While these provide good coverage of open and proprietary systems, additional testing on other models (e.g., LLaMA, Mistral, Claude) would better establish generality across architectures and instruction styles.

References

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, et al. 2024. Longbench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3119–3137.

Federico Barbero, Andrea Banino, Steven Kapturowski, Dharshan Kumaran, João Madeira Araújo, Oleksandr Vitvitskyi, Razvan Pascanu, and Petar Veličković. 2024. Transformers need glasses! information over-squashing in language tasks. *Advances in Neural Information Processing Systems*, 37:98111–98142.

Yelysei Bondarenko, Markus Nagel, and Tijmen Blankevoort. 2021. Understanding and overcoming the challenges of efficient transformer quantization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7947–7969. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Zhiwei Cao, Qian Cao, Yu Lu, Ningxin Peng, Luyang Huang, Shanbo Cheng, and Jinsong Su. 2024. Retaining key information under high compression ratios: Query-guided compressor for LLMs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12685–12695.

Chris Ding and Hanchuan Peng. 2005. Minimum redundancy feature selection from microarray gene expression data. *Journal of bioinformatics and computational biology*, 3(02):185–205.

Yixiong Fang, Tianran Sun, Yuling Shi, and Xiaodong Gu. 2025. Attentionrag: Attention-guided context pruning in retrieval-augmented generation. *arXiv preprint arXiv:2503.10720*.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *International conference on machine learning*, pages 3929–3938. PMLR.

Yanwen Huang, Yong Zhang, Ning Cheng, Zhitao Li, Shaojun Wang, and Jing Xiao. 2025. Dynamic attention-guided context decoding for mitigating context faithfulness hallucinations in large language models. *arXiv preprint arXiv:2501.01059*.

Taeho Hwang, Sukmin Cho, Soyeong Jeong, Hoyun Song, SeungYoon Han, and Jong C Park. 2024. Exit: Context-aware extractive compression for enhancing retrieval-augmented generation. *arXiv preprint arXiv:2412.12559*.

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2023. LLMLingua: Compressing prompts for accelerated inference of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13358–13376.

Huiqiang Jiang, Qianhui Wu, Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2024a. LongLLMLingua: Accelerating and enhancing LLMs in long context scenarios via prompt compression. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1658–1677.

Ting Jiang, Shaohan Huang, Zhongzhi Luan, Deqing Wang, and Fuzhen Zhuang. 2024b. Scaling sentence embeddings with large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 3182–3196.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Yucheng Li, Bo Dong, Frank Guerin, and Chenghua Lin. 2023. Compressing context to enhance inference efficiency of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6342–6353.

Zhonghao Li, Xuming Hu, Aiwei Liu, Kening Zheng, Sirui Huang, and Hui Xiong. 2024. *Refiner*: Restructure retrieved content efficiently to advance question-answering capabilities. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 8548–8572.

Barys Liskavets, Maxim Ushakov, Shuvendu Roy, Mark Klibanov, Ali Etemad, and Shane Luke. 2024. Prompt compression with context-aware sentence encoding for fast and improved llm inference. *arXiv preprint arXiv:2409.01227*.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in gpt. *Advances in Neural Information Processing Systems*, 35:17359–17372.

OpenAI. 2024. Gpt-4 technical report.

Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Menglin Xia, Xufang Luo, Jue Zhang, Qingwei Lin, Victor Rühle, Yuqing Yang, Chin-Yew Lin, H. Vicky Zhao, Lili Qiu, and Dongmei Zhang. 2024. LLMLingua-2: Data distillation for efficient and faithful task-agnostic prompt compression. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 963–981.

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Richard James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2024. REPLUG: Retrieval-augmented black-box language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8371–8384.

Seungwoo Son, Wonpyo Park, Woohyun Han, Kyuyeun Kim, and Jaeho Lee. 2024. Prefixing attention sinks can mitigate activation outliers for large language model quantization. *arXiv preprint arXiv:2406.12016*.

Wenshan Wang, Yihang Wang, Yixing Fan, Huaming Liao, and Jiafeng Guo. 2024. Quito: Accelerating long-context reasoning through query-guided context compression. *arXiv preprint arXiv:2408.00274*.

Wenhao Wu, Yizhong Wang, Guangxuan Xiao, Hao Peng, and Yao Fu. 2024. Retrieval head mechanistically explains long-context factuality. *arXiv preprint arXiv:2404.15574*.

Fangyuan Xu, Weijia Shi, and Eunsol Choi. 2024. RECOMP: Improving retrieval-augmented LMs with context compression and selective augmentation. In *The Twelfth International Conference on Learning Representations*.

Ori Yoran, Tomer Wolfson, Ori Ram, and Jonathan Berant. 2024. Making retrieval-augmented language models robust to irrelevant context. In *The Twelfth International Conference on Learning Representations*.

Qianchi Zhang, Hainan Zhang, Liang Pang, Hongwei Zheng, Yongxin Tong, and Zhiming Zheng. 2025. Finefilter: A fine-grained noise filtering mechanism for retrieval-augmented large language models. *arXiv preprint arXiv:2502.11811*.

A Dataset Details

We provide a detailed description of the datasets used in our experiments, based on the English subset of LongBench (Bai et al., 2024). LongBench is a long-context benchmark covering diverse tasks designed to evaluate the capabilities of language models in understanding and reasoning over extended textual inputs. It consists of six task categories, each comprising multiple representative datasets:

- **Single-Document QA**:
  - NarrativeQA: Answer questions based on a single narrative document, such as a story or movie script.
  - Qasper: Answer questions grounded in a scientific paper.
  - MultiFieldQA-en: Answer factual questions from a long structured encyclopedic entry.
- **Multi-Document QA**:
  - HotpotQA, 2WikiMultihopQA, and MuSiQue: Multi-hop QA tasks requiring reasoning across multiple passages to answer complex factoid questions.
- **Summarization**:
  - GovReport: Summarize long government reports.

  - QMSUM: Summarize meeting transcripts.
  - MULTINEWS: Summarize multi-source news articles.

- **Few-shot Reasoning**:
  - TREC: Classify question types.
  - TRIVIAQA: Answer trivia-style factual questions.
  - SAMSUM: Summarize short dialogues.
  - LSHT: Classify Chinese news headlines into topic categories.

- **Synthetic Retrieval**:
  - PASSAGECOUNT: Count the number of unique paragraphs among potentially duplicated inputs.
  - PASSAGERETRIEVAL-EN: Identify the source paragraph corresponding to a given abstract.

- **Code Completion**:
  - LCC: Predict the next line of code given a code block, without an explicit natural language query.
  - REPOBENCH-P: Predict the next line of a function given multi-file code context and the function signature.

Dataset Filtering. We exclude two tasks—LCC and PASSAGECOUNT—from our evaluation due to incompatibility with query-conditioned compression. LCC lacks an explicit query, requiring the model to complete the final line of code based solely on preceding lines. Without a query to anchor attention, our method may prune essential lines that appear semantically uninformative. While this can potentially be addressed by treating the instruction "Next line of code" as a synthetic query, we leave this for future work. The PASSAGECOUNT task involves counting exact duplicates, which is incompatible with lossy compression: small differences between seemingly redundant paragraphs can lead to incorrect counts.

B Baseline Descriptions

We compare Sentinel against the following baseline methods, grouped by their design paradigms:

- **LLMLingua-1/2** (Jiang et al., 2023; Pan et al., 2024): Token-level compression methods based on saliency estimation via perplexity and LLM distillation. These methods are task-agnostic and do not condition on the query.

- **Selective-Context** (Li et al., 2023): A sentence-level, task-agnostic method that scores context segments based on general informativeness, independent of the question.

- **LongLLMLingua** (Jiang et al., 2024a): A query-aware, multi-stage compression system using query-conditioned perplexity scoring, document reordering, and adaptive compression ratios.

- **CPC** (Liskavets et al., 2024): A contrastively trained sentence-ranking model that selects sentences based on semantic similarity to the query in embedding space. It is query-aware and trained on synthetic QA data.

- **Raw Attention** (Wang et al., 2024; Fang et al., 2025): A non-learning baseline that selects sentences by averaging attention weights from the final decoder token. This mimics attention-based heuristics used in prior work such as QUITO and AttentionRAG.

- **Random Selection**: Sentences are sampled uniformly at random until the token budget is met. Serves as a lower-bound reference.

- **Empty Context**: The model receives only the question without any retrieved context, serving as a zero-context baseline.

All baselines are evaluated under the same token budget and LLM generation setting for fair comparison.

C mRMR Feature Selection Details

To construct a compact attention-based feature set, we use the Minimum Redundancy Maximum Relevance (mRMR) algorithm. We first compute mutual information between each feature (i.e., attention head statistics) and the binary relevance label, selecting the most informative one. We then iteratively add features that maximize relevance while minimizing redundancy, measured via Pearson correlation with already selected features. The number of features is capped at the number of heads in a single decoder layer to ensure compactness and interpretability.

D Chunk Size and Compression Ratio Details

We provide a subset of Table 2 to highlight the effect of chunk size and compression ratio in our ablation studies. The following tables report detailed task-level performance using the 0.5B proxy model.

Chunk Size. Table 5 reports results with different chunk sizes (512 to 4096 tokens), under a fixed 2000-token constraint. Despite being trained on short contexts, the model benefits from longer chunks at inference, suggesting gains from preserving broader intra-chunk coherence.

Compression Ratio. Table 6 reports results with varying compression ratios ($\tau \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$), under a fixed chunk size of 1024. Sentinel remains robust even at high compression, while Raw attention deteriorates significantly.

E Additional Chinese Results with GPT-3.5-Turbo

To assess the cross-lingual robustness of our method, we evaluate Sentinel on LongBench-Zh using GPT-3.5-Turbo as the inference model. We compare against LLMLingua and LLMLingua-2 baselines, which are evaluated under a 3,000-token input constraint. Sentinel uses only 2,000 tokens but consistently outperforms the baselines across all task categories.

F Sentence-Level Overlap Across Proxy Models

To examine the consistency of attention-based relevance signals across proxy model scales, we compute the pairwise sentence-level overlap of selected sentences. Figure 3 shows the overlap heatmaps between the 0.5B, 1.5B, and 3B models at different token budgets (1000, 2000, and 3000). Token-level overlap increases with budget, ranging from 0.63–0.70 at 1000 tokens to 0.74–0.78 at 3000 tokens, suggesting consistent relevance estimation across proxy model scales.

G t-SNE Visualization of Probing Features

To investigate how attention-derived sentence features vary across datasets and proxy model sizes, we visualize probing features using t-SNE. Each point represents a sentence-level example (either positive or negative), based on decoder attention from different proxy models: `Qwen-2.5-0.5B-Instruct`, `1.5B`, and `3B`. Visualizations are generated from 6,000 samples across SQuAD, NewsQA, and HotpotQA.

Figure 4a shows the feature space from the 0.5B proxy. We observe that positive and negative examples in SQuAD and NewsQA form distinguishable clusters, while HotpotQA features are more entangled—likely due to the diffuse nature of multi-hop supervision. This aligns with our observation that multi-hop sentence relevance is harder to learn from attention alone.

Figures 4b and 4c depict the same projection under larger proxies. Notably, the overall structure remains consistent, supporting our hypothesis that attention-based relevance behavior is stable across model scales.

H Additional Results: Training Size Ablation

Effect of Probing Data Size. We evaluate how training size affects probing quality. As shown in Table 8, performance remains stable across 500–3000 training examples, with only marginal gains. This suggests that even a small probing set can support effective compression.

I Attention Visualization Examples

We provide qualitative visualizations of two adjacent chunks from a `2WikiMultihopQA` example, processed by the `Qwen-2.5-0.5B-Instruct` proxy model with chunk size 1024. Each visualization illustrates three relevance estimation strategies:

- **Top:** Token-level attention weights from the decoder's final token.
- **Middle:** Averaged sentence attention, computed by averaging token-level attention over each sentence (Raw Attention baseline).
- **Bottom:** Sentence-level relevance predictions from our probing-based classifier.

As shown in Figures 5 and 6, token-level attention displays strong sink behavior, with most weights concentrated on the final input token. Sentence-level averaging slightly reduces noise but remains sensitive to this sink effect and lacks semantic alignment. In contrast, Sentinel's probing

Methods	LongBench-En (Qwen-2.5-7B-Instruct, 2000-token constraint)							LongBench-Zh (Qwen-2.5-7B-Instruct, 2000-token constraint)						Overall AVG
	SingleDoc	MultiDoc	Summ.	FewShot	*Synth.	*Code	En-AVG	SingleDoc	MultiDoc	Summ.	FewShot	Synth.	Zh-AVG	
Raw Attention (512)	31.67	36.01	24.19	64.37	72.92	62.93	48.68	46.50	19.32	12.77	39.25	55.90	34.75	45.70
Raw Attention (1024)	33.11	38.39	24.22	63.27	77.67	62.14	49.80	56.24	20.99	13.61	44.50	72.76	41.62	48.47
Raw Attention (2048)	34.51	36.54	24.17	65.21	77.50	61.35	49.88	56.17	23.14	13.36	40.50	75.39	41.71	48.63
Raw Attention (4096)	34.79	41.17	24.20	65.59	77.08	62.46	50.88	56.90	23.94	13.42	43.50	75.70	42.69	49.66
Sentinel (512)	36.97	40.60	24.73	64.80	78.42	63.71	51.54	57.88	23.17	13.50	42.50	68.40	41.09	49.51
Sentinel (1024)	37.11	44.98	25.02	64.88	85.54	63.04	53.43	59.18	24.15	13.19	43.00	74.68	42.84	51.23
Sentinel (2048)	36.15	43.20	25.09	65.03	90.50	61.67	53.60	61.41	25.07	13.06	42.50	80.26	44.46	51.68
Sentinel (4096)	37.79	44.36	25.13	65.88	88.33	61.66	53.86	59.07	25.14	12.81	41.50	81.12	43.93	51.87

Table 5: Performance across chunk sizes with a fixed 2000-token constraint.

Methods	LongBench-En (Qwen-2.5-7B-Instruct, chunk size = 1024)							LongBench-Zh (Qwen-2.5-7B-Instruct, chunk size = 1024)						Overall AVG
	SingleDoc	MultiDoc	Summ.	FewShot	*Synth.	*Code	En-AVG	SingleDoc	MultiDoc	Summ.	FewShot	Synth.	Zh-AVG	
Raw Attention (ratio 0.1)	24.83	33.73	21.84	60.56	63.67	58.66	43.88	33.17	18.95	12.65	34.77	31.82	26.27	39.52
Raw Attention (ratio 0.2)	30.88	36.64	23.21	64.61	81.33	60.57	47.19	45.43	20.90	13.33	42.82	49.60	32.88	43.62
Raw Attention (ratio 0.3)	32.50	39.85	24.04	64.93	90.50	61.62	52.24	50.97	20.05	14.16	39.81	61.84	37.37	48.39
Raw Attention (ratio 0.4)	34.85	40.49	24.35	66.10	94.50	61.81	53.68	54.50	19.77	14.11	41.61	70.70	40.14	50.26
Raw Attention (ratio 0.5)	35.44	38.87	24.86	67.54	94.83	62.48	54.00	57.33	20.52	14.31	45.13	78.51	43.16	51.48
Sentinel (ratio 0.1)	34.28	38.14	22.93	60.69	75.04	58.86	48.33	56.91	22.89	13.06	38.64	56.04	37.51	45.78
Sentinel (ratio 0.2)	36.22	42.70	24.17	64.72	85.17	61.08	52.34	58.71	21.68	13.84	42.84	71.51	41.72	49.99
Sentinel (ratio 0.3)	36.79	42.42	24.66	66.72	92.00	62.11	54.12	58.05	22.37	14.31	41.02	77.78	42.71	51.54
Sentinel (ratio 0.4)	37.31	41.35	24.89	66.82	92.92	61.71	54.17	57.66	22.40	14.31	43.55	83.51	44.29	52.09
Sentinel (ratio 0.5)	38.22	40.98	24.90	65.56	94.75	61.58	54.10	59.21	21.18	14.68	43.40	83.59	44.41	51.99

Table 6: Performance across compression ratios (chunk size = 1024).

classifier produces sparse and interpretable predictions that reliably highlight answer-supporting sentences across chunks.

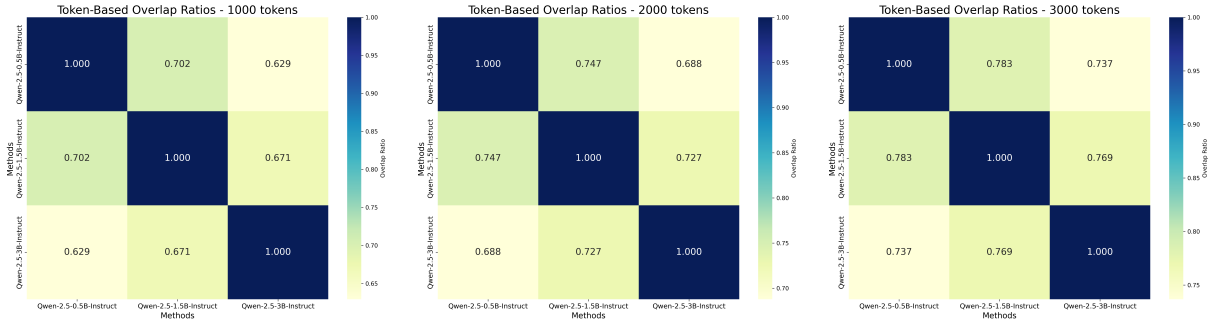
J LLM Evaluation Settings

For LLM-based evaluation, we adopt the official prompt templates and decoding settings used in LongBench (Bai et al., 2024) to ensure consistency and comparability. The decoding parameters are fixed across all datasets as follows:

- `temperature`: 0.0
- `top_p`: 1.0
- `seed`: 42
- `n`: 1
- `stream`: False
- `max_tokens`: dataset-specific (see Table 9)

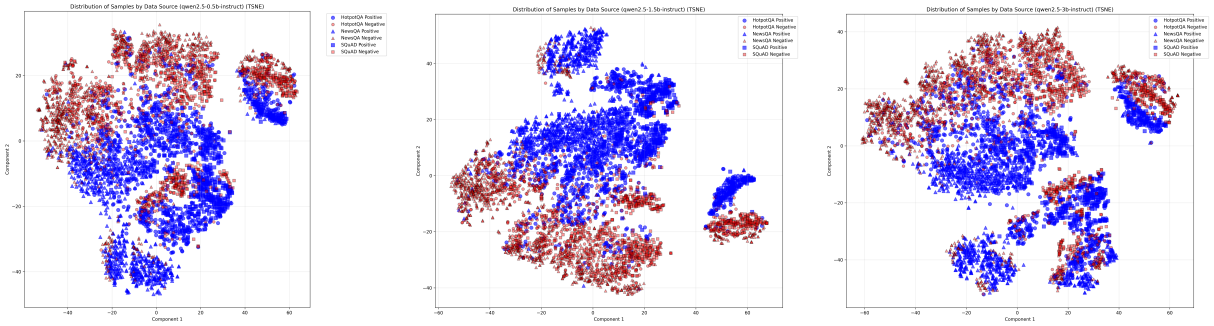
Methods	LongBench-Zh (GPT-3.5-Turbo, 3000-token constraint)						Compression Stats	
	SingleDoc	MultiDoc	Summ.	FewShot	Synth.	**AVG**	Tokens	$1/\tau$
LLMLingua	35.2	20.4	11.8	24.3	51.4	28.6	3,060	5x
LLMLingua-2	46.7	23.0	15.3	32.8	**72.6**	38.1	3,023	5x
Evaluated under 2000-token constraint								
Sentinel (Qwen-2.5-0.5B-Instruct)	**64.8**	25.1	14.3	38.0	89.0	46.2	1,932	5x
Sentinel (Qwen-2.5-1.5B-Instruct)	63.3	24.9	14.8	**40.3**	**95.0**	**47.6**	1,929	5x
Original Prompt	61.2	**28.7**	**16.0**	29.2	77.5	42.5	14,940	-

Table 7: Performance comparison on LongBench-Zh using GPT-3.5-Turbo. LLMLingua baselines are evaluated under a 3,000-token budget. Sentinel uses only 2,000 tokens but consistently outperforms the baselines, demonstrating effective compression across languages.



(a) Token budget = 1000 (b) Token budget = 2000 (c) Token budget = 3000

Figure 3: Pairwise sentence-level overlap between proxy models at different token budgets. Higher overlap indicates stronger alignment in relevance estimation.



(a) `Qwen-2.5-0.5B-Instruct` (b) `Qwen-2.5-1.5B-Instruct` (c) `Qwen-2.5-3B-Instruct`

Figure 4: t-SNE visualization of probing features across three proxy model scales. Each point represents a sentence from SQuAD, NewsQA, or HotpotQA. Despite model size differences, the distributional structure remains stable—supporting scale-invariant attention behavior.

Methods	LongBench-En (Qwen-2.5-7B-Instruct, 2000-token constraint)							LongBench-Zh (Qwen-2.5-7B-Instruct, 2000-token constraint)						Overall AVG
	SingleDoc	MultiDoc	Summ.	FewShot	*Synth.	*Code	En-AVG	SingleDoc	MultiDoc	Summ.	FewShot	Synth.	Zh-AVG	
Qwen-2.5-0.5B-Instruct (500)	36.42	44.54	24.99	66.01	89.58	63.12	54.11	60.60	24.52	13.22	43.25	75.42	43.40	51.73
Qwen-2.5-0.5B-Instruct (1000)	36.11	44.50	24.95	66.06	88.75	62.10	53.74	60.13	23.53	13.38	42.50	77.69	43.45	51.55
Qwen-2.5-0.5B-Instruct (2000)	36.67	44.58	24.88	64.52	87.04	62.91	53.43	60.55	25.42	13.11	41.25	74.13	42.89	51.16
Qwen-2.5-0.5B-Instruct (3000)	37.11	44.98	25.02	64.88	85.54	63.04	53.43	59.18	24.15	13.19	43.00	74.68	42.84	51.23

Table 8: Performance of 0.5B models with different probing sizes (500, 1000, 2000, 3000) on LongBench.

Figure 5: Visualization of the second chunk in a 2WikiMultihopQA example. Top: token-level attention weights from the final decoder token. Middle: sentence-level scores from averaged token attention (Raw Attention baseline). Bottom: sentence-level relevance predictions from the probing-based classifier.

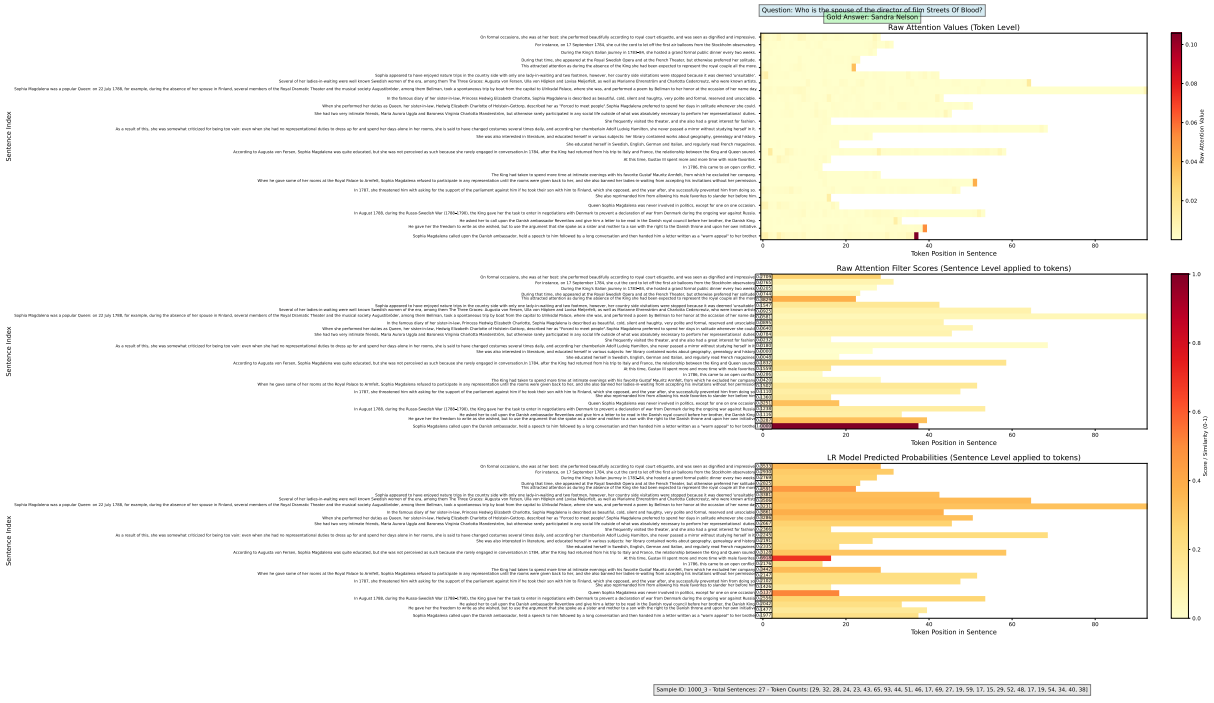


Figure 6: Visualization of the third chunk in a 2WikiMultihopQA example. Top: token-level attention weights from the final decoder token. Middle: sentence-level scores from averaged token attention (Raw Attention baseline). Bottom: sentence-level relevance predictions from the probing-based classifier.

Dataset	Max Tokens
narrativeqa	128
qasper	128
multifieldqa_en	64
multifieldqa_zh	64
hotpotqa	32
2wikimqa	32
musique	32
dureader	128
gov_report	512
qmsum	512
multi_news	512
vcsum	512
trec	64
triviaqa	32
samsum	128
lsht	64
passage_count	32
passage_retrieval_en	32
passage_retrieval_zh	32
lcc	64
repobench-p	64

Table 9: Maximum number of generation tokens for each dataset.